

EXHIBIT B

```

package KnowledgeAgents;

import se.*;
import gui.*;
5  import utils.*;

import java.awt.*;
import java.awt.event.*;
import java.util.Stack;
10 import java.io.*;

/**
 * The user interface of the Agent.
 */
15 public class AgentGUI extends Frame
    implements ActionListener, ItemListener
{
    Agent agent;
    Trace trace;

20    /* private gui instance variables */
    CheckboxGroup    queryType;
    Checkbox          textQuery, linkQuery, repSites;
    private TextField tfQuery;

25    CheckboxGroup    engineOptions;
    Checkbox          allEngines, selectEngines;
    List              engines;

30    Checkbox          fileSelect;
    private TextField tfOutputFile;

    private TextField tfExpandFactor, tfRootSetSize, tfNumResults;
    private TextField tfMaxLinkWeight, tfQueryRefineFactor;

35    private TextField tfNSites, tfPhase, tfLeft;
    private TextField tfServQIn, tfServQOut, tfServQUtil;
    private boolean    vis;
    private Button     queryRefine, addSites;
40    private Button     searchLearn, searchOnly;

    /* public methods */

    /** */
45    public AgentGUI(Agent agent,
        Trace trace )
    {
        this.agent = agent;
        this.trace = trace;

50        int    gridy = 0, gridx = 1;
        Font    fntTitle = new Font("Helvetica", Font.ITALIC, 18);
        Font    fnt      = new Font("TimesRoman", Font.PLAIN, 12);
        Font    fntBold  = new Font("TimesRoman", Font.BOLD, 12);

55        Panel          p          = new Panel();
        ScrollPane       scroll     = new ScrollPane();
        GridBagLayout    gbl       = new GridBagLayout();
        GridBagConstraints c       = new GridBagConstraints();

60        vis = true;

```

```

p.setFont (fnt);
p.setLayout(gbl);

65 // prepare the panel's title
/*
GuiAddFields.AddLabel ("Knowledge Agent Window",
                        true,
70                        0, gridy++, 4, 1,
                        GridBagConstraints.CENTER,
                        fntTitle,
                        gbl, p );
GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);
75 GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);
*/
//*****

GuiAddFields.AddLabel
80 ("Enter Query/Add Repository Sites",
    false,
    0, gridy++, 2, 1,
    GridBagConstraints.CENTER,
    fntBold,
85    gbl, p );

// add the radio buttons and the file dialog
queryType = new CheckboxGroup();
textQuery = new Checkbox ("Text query", queryType, true);
90 linkQuery = new Checkbox ("Sample URL query", queryType, false);
repSites = new Checkbox ("Repository Sites", queryType, false);

GuiAddFields.AddCheckbox(textQuery, 0, gridy, 1,1,gbl,p,(Frame)this);
tfQuery = GuiAddFields.AddTextField
95 (45,
    1, gridy++, 4, 1,
    GridBagConstraints.WEST,
    false,
    gbl, p );
100 tfQuery.setEditable(true);
GuiAddFields.AddCheckbox(linkQuery, 0, gridy, 1,1,gbl,p,(Frame)this);
GuiAddFields.AddCheckbox(repSites, 0, gridy+1, 1,1,gbl,p,(Frame)this);

GuiAddFields.AddLabel (" Query Refinement Factor:",
105 false,
    1, gridy, 2, 1,
    GridBagConstraints.CENTER,
    fnt,
    gbl, p );
110 tfQueryRefineFactor= GuiAddFields.AddTextField
    (5,
    3, gridy, 1, 1,
    GridBagConstraints.WEST,
    false,
115    gbl, p );
tfQueryRefineFactor.setText("2");
queryRefine = GuiAddFields.AddButton ("Refine Query", 4, gridy++,
    GridBagConstraints.CENTER,
    gbl, p, (Frame)this);
120 addSites = GuiAddFields.AddButton ("Add Sites", 1, gridy++,
    GridBagConstraints.WEST,
    gbl, p, (Frame)this);

```

```

GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);
125 //*****
GuiAddFields.AddLabel
    ("Choose Search Engines:",
     false,
     0, gridy, 2, 1,
130     GridBagConstraints.CENTER,
     fntBold,
     gbl, p
     );
GuiAddFields.AddLabel
    ("Various Arguments:",
135     false,
     3, gridy++, 2, 1,
     GridBagConstraints.CENTER,
     fntBold,
     gbl, p
     );
140
// add the radio buttons and the scrollable list
engineOptions = new CheckboxGroup();
allEngines     = new Checkbox ("All Engines", engineOptions, false);
selectEngines = new Checkbox ("Select From List --> ",
145     engineOptions, true
    );

GuiAddFields.AddCheckbox(allEngines,
    0, gridy, 1,1,gbl,p,(Frame)this);
GuiAddFields.AddCheckbox(selectEngines,
150     0, gridy+1, 1,1,gbl,p,(Frame)this);

engines = new List(SearchEngine.NUM_ENGINES,true);
for ( int i=0; i<SearchEngine.NUM_ENGINES; i++ )
    engines.addItem(SearchEngine.engineNames[i]);
155 GuiAddFields.AddComponent (engines, 1, gridy,1,2,gbl,p);

GuiAddFields.AddLabel ("Max. Link Weight:",
    false,
    3, gridy, 2, 1,
160     GridBagConstraints.CENTER,
    fnt,
    gbl, p
    );
tfMaxLinkWeight = GuiAddFields.AddTextField
    (5,
165     5, gridy, 1, 1,
     GridBagConstraints.WEST,
     false,
     gbl, p
     );
tfMaxLinkWeight.setText("1.25");
170

gridy+=2;
GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);

//*****
175
GuiAddFields.AddLabel
    ("Set Size Restrictions",
     false,
     0, gridy++, 2, 1,
180     GridBagConstraints.CENTER,
     fntBold,
     gbl, p
     );

```

```

185     GuiAddFields.AddLabel ("Root Set Size:",
                             false,
                             0, gridy, 1, 1,
                             GridBagConstraints.CENTER,
                             fnt,
                             gbl, p
                             );
190     tfRootSetSize = GuiAddFields.AddTextField
                             (3,
                             1, gridy, 1, 1,
                             GridBagConstraints.WEST,
                             false,
195     tfRootSetSize.setText("30");

     GuiAddFields.AddLabel ("Expand Factor:",
                             false,
200     2, gridy, 1, 1,
                             GridBagConstraints.CENTER,
                             fnt,
                             gbl, p
                             );
     tfExpandFactor = GuiAddFields.AddTextField
205     (3,
     3, gridy, 1, 1,
     GridBagConstraints.WEST,
     false,
     gbl, p
     );
210     tfExpandFactor.setText("40");

     GuiAddFields.AddLabel ("Show Best:",
                             false,
215     4, gridy, 1, 1,
                             GridBagConstraints.CENTER,
                             fnt,
                             gbl, p
                             );
     tfNumResults = GuiAddFields.AddTextField
220     (3,
     5, gridy++, 1, 1,
     GridBagConstraints.WEST,
     false,
     gbl, p
     );
     tfNumResults.setText("15");
225     GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);

    //*****

230     GuiAddFields.AddLabel
                             ("Execution Trace",
                             false,
                             0, gridy++, 1, 1,
                             GridBagConstraints.CENTER,
235     fntBold,
                             gbl, p
                             );

     GuiAddFields.AddLabel ("Number Of Sites:",
                             false,
240     0, gridy, 1, 1,
                             GridBagConstraints.CENTER,
                             fnt,
                             gbl, p
                             );
     tfNSites = GuiAddFields.AddTextField

```

```

245             (5,
                1, gridy, 1, 1,
                GridBagConstraints.WEST,
                false,
                gbl, p            );
250 tfNSites.setText("0");
    tfNSites.setEditable(false);

    GuiAddFields.AddLabel ("URLs Left:",
                            false,
255                2, gridy, 1, 1,
                GridBagConstraints.CENTER,
                fnt,
                gbl, p            );
    tfLeft = GuiAddFields.AddTextField
260                (4,
                3, gridy, 1, 1,
                GridBagConstraints.CENTER,
                false,
                gbl, p            );
265 tfLeft.setText("");
    tfLeft.setEditable(false);

    GuiAddFields.AddLabel
                            ("Dedicated Rdrs:",
270                false,
                4, gridy, 1, 1,
                GridBagConstraints.CENTER,
                fnt,
                gbl, p            );
275 tfServQUtil= GuiAddFields.AddTextField
                (3,
                5, gridy++, 1, 1,
                GridBagConstraints.WEST,
                false,
280                gbl, p            );
    tfServQUtil.setText("");
    tfServQUtil.setEditable(false);

    GuiAddFields.AddLabel ("Phase:",
285                false,
                0, gridy, 1, 1,
                GridBagConstraints.CENTER,
                fnt,
                gbl, p            );
290 tfPhase = GuiAddFields.AddTextField
                (35,
                1, gridy, 3, 1,
                GridBagConstraints.WEST,
                false,
295                gbl, p            );
    tfPhase.setText("");
    tfPhase.setEditable(false);

    tfServQIn = GuiAddFields.AddTextField
300                (3,
                4, gridy, 1, 1,
                GridBagConstraints.WEST,
                false,
                gbl, p            );
305 tfServQIn.setText("");

```

```

tfServQIn.setEditable(false);
tfServQOut = GuiAddFields.AddTextField
    (3,
    5, gridy++, 1, 1,
310      GridBagConstraints.WEST,
      false,
      gbl, p
    );
tfServQOut.setText("");
tfServQOut.setEditable(false);
315 GuiAddFields.AddLabel
    ("Pending Read Reqs/Ans",
    false,
    4, gridy++, 2, 1,
    GridBagConstraints.CENTER,
320      fnt,
      gbl, p
    );

//*****

325 GuiAddFields.AddLabel
    ("Choose Output File:",
    false,
    0, gridy++, 1, 1,
    GridBagConstraints.CENTER,
330      fntBold,
      gbl, p
    );
GuiAddFields.AddButton ("Select File...", 0, gridy,
    GridBagConstraints.WEST,
    gbl, p, (Frame)this);
335

tfOutputFile = GuiAddFields.AddTextField
    (45,
    1, gridy, 4, 1,
    GridBagConstraints.WEST,
340      false,
      gbl, p
    );
tfOutputFile.setEditable(true);

GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);
345

//*****

GuiAddFields.AddButton ("Trace...", 0, gridy, GridBagConstraints.WEST,
    gbl, p, (Frame)this);
350 GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);

//*****

GuiAddFields.AddButton ("Save State", 0, gridy,
    GridBagConstraints.CENTER,
    gbl, p, (Frame)this);
355 GuiAddFields.AddButton ("Restore State", 1, gridy,
    GridBagConstraints.CENTER,
    gbl, p, (Frame)this);

360 searchLearn =
GuiAddFields.AddButton ("Search & Learn", 3, gridy,
    GridBagConstraints.CENTER,
    gbl, p, (Frame)this);

searchOnly =
365 GuiAddFields.AddButton ("Search Only", 5, gridy,
    GridBagConstraints.CENTER,

```

```

        gbl, p, (Frame)this);

    gridy+=2;
370  GuiAddFields.AddEmptyLine (gridy++, fnt, gbl, p);
    //*****

    GuiAddFields.AddButton ("Exit Agent", 2, gridy,
375        GridBagConstraints.CENTER,
        gbl, p, (Frame)this);

    //*****

    /* general settings */
380  setBackground(Color.gray);
    addWindowListener(new DWAdapter());

    scroll.add(p);
    add("Center", scroll);
385  add("West", new Label("  "));
    add("East", new Label("  "));
    add("South", new Label("  "));
}

390  public void itemStateChanged(ItemEvent event)
    {
        if (event.getItemSelectable() == allEngines)
        {
            engines.setVisible(false);
395        }
        else if (event.getItemSelectable() == selectEngines)
        {
            engines.setVisible(true);
        }
400        else if (event.getItemSelectable() == textQuery)
        {
            tfQuery.setEditable(true);
            tfRootSetSize.setVisible(true);
            tfQueryRefineFactor.setVisible(true);
405            queryRefine.setVisible(true);
            searchLearn.setVisible(true);
            searchOnly.setVisible(true);
            addSites.setVisible(false);
        }
410        else if (event.getItemSelectable() == linkQuery)
        {
            tfQuery.setEditable(true);
            tfRootSetSize.setVisible(false);
            tfQueryRefineFactor.setVisible(false);
415            queryRefine.setVisible(false);
            searchLearn.setVisible(true);
            searchOnly.setVisible(true);
            addSites.setVisible(false);
        }
420        else if (event.getItemSelectable() == repSites)
        {
            tfQuery.setEditable(true);
            tfQueryRefineFactor.setVisible(false);
            queryRefine.setVisible(false);
425            searchLearn.setVisible(false);
            searchOnly.setVisible(false);
            addSites.setVisible(true);
        }
    }

```

```

    }
}
430

public void actionPerformed(ActionEvent event)
{
    String command = event.getActionCommand();

435    if ( command == "Refine Query" )
    {
        Agent.Action action = (Agent.Action) new Agent.ActionRefineQuery
            (tfQuery.getText(),
440            new Integer (tfQueryRefineFactor.getText()).intValue(),
            tfQuery);
        agent.setAction(action);
    }
    else
445    if ( command == "Add Sites" )
    {
        Agent.Action action = (Agent.Action) new Agent.ActionAddSites
            (tfQuery.getText());
        agent.setAction(action);
450    }
    else
    if (command == "Select File..." )
    {
        FileDialog fd = new FileDialog(this, "Agent File");
455        fd.show();
        if (fd.getDirectory() != null && fd.getFile() != null )
            tfOutputFile.setText(fd.getDirectory()+fd.getFile());
    }
    else
460    if (command == "Trace..." )
    {
        TraceDialog td = new TraceDialog(this,
            "Agent Trace Options",
            Trace.AGENT_TRACE,
465            trace        );
        td.show();
    }
    else
    if (command == "Save State")
470    {
        FileDialog fd = new FileDialog(this, "Agent File");
        fd.show();
        if (fd.getDirectory() != null && fd.getFile() != null )
        {
475            String fileName = fd.getDirectory()+fd.getFile();
            try
            {
                ObjectOutputStream oos =
                    new ObjectOutputStream(new FileOutputStream(fileName));
480                agent.saveState(oos);
                oos.flush();
                oos.close();
            }
            catch (Exception e)
485            {
                System.err.println("Error while saving the Agent's state:\n"+e);
            }
        }
    }
}

```



```

    }
490 else
    if (command == "Restore State")
    {
        FileDialog fd = new FileDialog(this, "Agent File");
        fd.show();
495     if (fd.getDirectory() != null && fd.getFile() != null )
        {
            String fileName = fd.getDirectory()+fd.getFile();
            try
            {
200         ObjectInputStream ois =
                new ObjectInputStream(new FileInputStream(fileName));
                agent.restoreState(ois);
                ois.close();
            }
505         catch (Exception e)
            {
                System.err.println("Error while saving the Agent's state:\n"+e);
            }
        }
510     }
    else
    if ( command == "Exit Agent")
    {
        Agent.Action action = new Agent.ActionExit();
515     agent.setAction(action);
    }
    else
    if (command == "Search & Learn" || command == "Search Only")
    {
520     boolean update = command == "Search & Learn" ? true : false;
        boolean goodInput = true;
        SearchEngine[] searchEngines = null;

        if ( tfOutputFile.getText().length() == 0 )
525     {
            StatusDialog sd = new StatusDialog
                (this,"Bad Input","Output File Must Be Chosen!");
            sd.show();
            goodInput = false;
        }
530     if ( textQuery.getState() &&
            tfRootSetSize.getText().length() == 0 )
        {
            StatusDialog sd = new StatusDialog
535     (this,"Bad Input","Size of Root Set Must Be Specified!");
            sd.show();
            goodInput = false;
        }
        if ( tfQuery.getText().length() == 0 )
540     {
            StatusDialog sd = new StatusDialog
                (this,"Bad Input","Query Must Be Specified!");
            sd.show();
            goodInput = false;
        }
545     if ( tfExpandFactor.getText().length() == 0 ||
            tfNumResults.getText().length() == 0 )
        {
            StatusDialog sd = new StatusDialog

```

```

550      (this,"Bad Input","Expand Factor/Num. Results Must Be Set!");
      sd.show();
      goodInput = false;
    }
    if ( allEngines.getState() )
555    {
      searchEngines = new SearchEngine [SearchEngine.NUM_ENGINES];
      try {
        // prepare an array of all search engines
        for ( int i = 0; i < SearchEngine.NUM_ENGINES; i++ )
560          searchEngines[i] = SEManager.ENGINES[i].seClone();
      }
      catch (CloneNotSupportedException e)
        { /* clone is supported by the Search Engine classes*/ }
    }
565    else
    {
      int[] selectedEngines = engines.getSelectedIndexes();

      if ( selectedEngines.length == 0 )
570      {
        StatusDialog sd = new StatusDialog
          (this,"Bad Input","Engines Must Be Chosen!");
        sd.show();
        goodInput = false;
575      }
      else
      {
        searchEngines = new SearchEngine[selectedEngines.length];
        try
580        {
          // prepare an array with clones of the selected engines
          for ( int i=0; i<selectedEngines.length; i++ )
            searchEngines[i]= (SearchEngine)
              SEManager.ENGINES[selectedEngines[i]].seClone();
585        }
        catch (CloneNotSupportedException e)
          { /* clone is supported by the Search Engine classes*/ }
      }
590    }

    if (goodInput)
    {
      Agent.Action action = textQuery.getState() ?
        (Agent.Action) new Agent.ActionTextQuery
595          (tfQuery.getText(),
            // Aya 20.10.99 - changed the paradigm, the user must refine
the query          // explicitly by clicking the refine query
button          // no automatic refinement during query execution
600          /* new Integer
            (tfQueryRefineFactor.getText()).intValue(), */
            0,
            new Integer (tfRootSetSize.getText()).intValue(),
605            new Integer (tfExpandFactor.getText()).intValue(),
            new Integer (tfNumResults.getText()).intValue(),
            new Float (tfMaxLinkWeight.getText()).floatValue(),
            tfOutputFile.getText(),
            searchEngines,
610            update) :

```

```

        (Agent.Action) new Agent.ActionLinkQuery
            (tfQuery.getText(),
             new Integer (tfExpandFactor.getText()).intValue(),
             new Integer (tfNumResults.getText()).intValue(),
615         new Float   (tfMaxLinkWeight.getText()).floatValue(),
             tfOutputFile.getText(),
             searchEngines,
             update);
        agent.setAction(action);
620     }
    }
    else
    {
        System.out.println(event);
625     }

    return;
}

630 public final void
    setPhase (String phase)
    {
        tfPhase.setText(phase);
    }
635 public final void
    setNSites (int n)
    {
        tfNSites.setText(String.valueOf(n));
    }
640 public final void
    setLeft (int n)
    {
        tfLeft.setText(String.valueOf(n));
    }
645 public final void
    setServQIn (int n)
    {
        tfServQIn.setText(String.valueOf(n));
    }
650 public final void
    setServQOut (int n)
    {
        tfServQOut.setText(String.valueOf(n));
    }
655 public final void
    setServQUtil (int n)
    {
        tfServQUtil.setText(String.valueOf(n));
    }
660 class DWAdapter extends WindowAdapter
    {
        public void windowClosing(WindowEvent event)
        {
            agent.exit();
665     }
    }
}

```